

An introduction to the UC Ergonomics Human Activity Recognition Deep Learning Project

by Yibin in Summer 2019

What is this

several UCL models that can classify human activities from Xsens sensor data. This project is based on a modified Residual Neural Network Model adapted from Resnet. It achieves 95% accuracy in the training/testing data and 90% accuracy in the task data prediction.

Package requirement

Pytorch, torchvision, and numpy. Recommend to install anaconda first.

Project structure

- Models **folder**: this folder contains all the model I built. See Model Description.txt for details about each model.
- Activity Recognition from Single Chest-Mounted Accelerometer **folder**: this folder contains the Activity Recognition from Single Chest-Mounted Accelerometer data from UCI. Only used this dataset as a proof-of-concept to my model.
- Spinetrack Data **folder**: all data that download from box folder. This folder has two sub-folder: data and task. data is used for model training, and the task folder is used for task prediction. See Data Description Section for more details.
- Files ending in **.ipynb**: these are the jupyter notebook that have all the code. See Jupyter Notebook Section for more details.
- force regression **folder**: not related to this project

Data Description

The whole Spinetrack Data folder is downloadable from box. data sub-folder includes all training data

(calibration), while task sub-folder contains all actual task data.

data:

each subject's calibration data is in this folder. For each person (except Niral), they all have three different data - raw data (directly under each person's folder), processed data (**folder** Processed), and further processed data (**folder** Processed-redone).

The .csv- files which are directly under the **data folder** are the raw .csv-files, meaning the non-processed output from the Matlab-app SpineTrackDataAcqv15.mlapp. In the *processed* as well as in the *processed_redone* folder the pauses in between the repetitions were removed using the *data_process_liftingv4.m* code. The difference between those folders is, that in the *processed_redone* folder also the starting portion of each lift was removed, whereas in *processed* strictly only the pauses were removed. We had problems with the deep learning model when part of the pauses were still in the data, that's why we created *processed_redone*.

task:

each person's actual activities data is in this folder. Unlike the **data folder** that repeats the same action in one .csv file, all activities in this folder are a sequence of activities that appears in the **data folder**. The model only use *everything.csv* as the prediction criteria.

Jupyter Notebook

Each jupyter notebook will automatically load the data and run either training the model or predicting the result.

Definition:

window_size: number of rows that combined together to form a *window*.

channel: default to 1, similar to channel in an image (RGB)

input: similar to an image (channel, width, height) input, the input of neural net has a tensor of size (channel, window_size, 19)

Files

- **Spinetrack Data Testing:** test the model with each person's *everything.csv* file. *everything.csv* contains a sequence of different task, and *timeStamps_Everything.txt* has a manual label of the activities. The model uses *everything.csv* as a task prediction criteria.

- **Spinetrack Data Testing - Loop Fast!:** similar to testing except it uses a loop to find the best accurate testing window.
- **Spinetrack Data Prediction:** predict the activities from *everything.csv*.
- **Spinetrack Data Training:** main training file. Trained from Spinetrack Data folder.
- **Spinetrack Sub-activities xxx Data Training:** train model from each sub-category activities. For example, in the main training, *carry10 and carry15* is the same activity as "carrying", but here depends on the file name it is responsible to train only one kind of activities.
- **Others:** not very related to this project or outdated.

Questions

If you have any question, feel free to contact Yibin at liyibin516@berkeley.edu.